Computing bubble oscillations on GPU

J. Jung, P. Helluy
Joint work with M. Bachmann, S. Müller

IRMA, University of Strasbourg

Paris, 7th DFG-CNRS workshop, February 15th, 2012

Outline

Model

We consider the Euler equations in 1D:

$$\begin{aligned}
\partial_t(\rho) + \partial_x(\rho u) &= 0, \\
\partial_t(\rho u) + \partial_x(\rho u^2 + p) &= 0, \\
\partial_t(\rho E) + \partial_x((\rho E + p)u) &= 0, \\
\partial_t(\rho \varphi) + \partial_x(\rho \varphi u) &= 0,
\end{aligned}$$

where ρ is the density, u the velocity, E the total energy, φ the fraction of mass of gas and p satisfies a mixture stiffened gas pressure law:

$$p(\rho, e, \varphi) = (\gamma(\varphi) - 1)\rho e - \gamma(\varphi)\pi(\varphi),$$

where

$$e = E - \frac{1}{2}u^2.$$

Mixture law

- We have

$$\partial_t \varphi + u \partial_x \varphi = 0$$

$\Rightarrow$ theoricaly φ is in $\{0, 1\}$ at any time.

- Numerically we have diffusion, we define a mixture pressure law with mixture parameters

$$\begin{aligned} \frac{1}{\gamma(\varphi) - 1} &= \varphi \frac{1}{\gamma_2 - 1} + (1 - \varphi) \frac{1}{\gamma_1 - 1}, \\ \frac{\gamma(\varphi)\pi(\varphi)}{\gamma(\varphi) - 1} &= \varphi \frac{\gamma_2 \pi_2}{\gamma_2 - 1} + (1 - \varphi) \frac{\gamma_1 \pi_1}{\gamma_1 - 1}. \end{aligned}$$

- We define the speed of sound

$$c = \sqrt{\gamma \frac{p + \pi}{\rho}}.$$

Model

- We can write the system in the conservative form:

$$\partial_t W + \partial_x F(W) = 0,$$

where the vector of conservative variables is:

$$W = (\rho, \rho u, \rho E, \rho\varphi)^T,$$

and the conservative flux is:

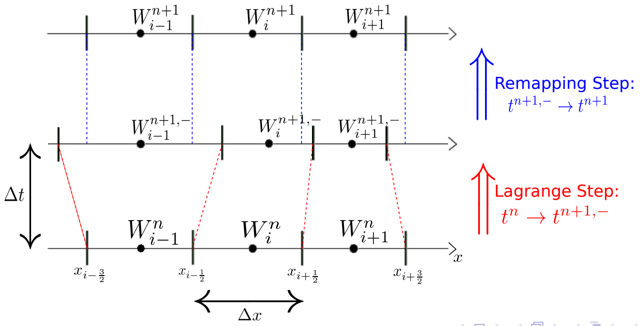
$$F(W) = (\rho u, \rho u^2 + p, (\rho E + p)u, \rho\varphi u)^T.$$

- This system is hyperbolic with the four eigenvalues $\lambda_1 = u - c$, $\lambda_2 = \lambda_3 = u$ and $\lambda_4 = u + c$.

Stucture of the scheme

The scheme includes two step:

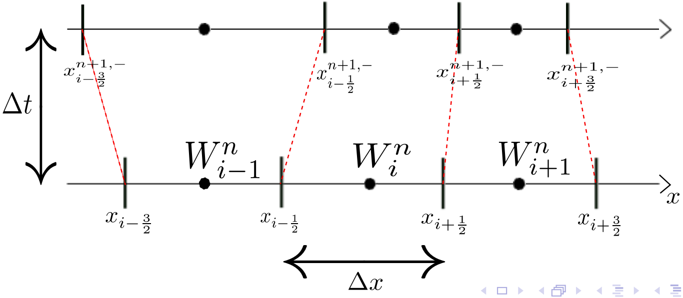
- the Lagrangian step to solve the system $\partial_t W + \partial_x F(W) = 0$, between time t^n and $t^{n+1,-}$,
- the remapping step to compute the Euler variable at time t^{n+1}.



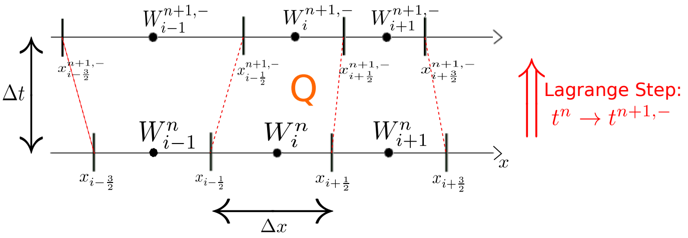
Some notation

- We propose a first order finite volume scheme with a Lagrangian approach, the boundary $x_{i+\frac{1}{2}}$ moves at the velocity of the fluid $u^n_{i+\frac{1}{2}}$ between t^n and $t^{n+1,-}$:

$$x^{n+1,-}_{i+\frac{1}{2}} = x^n_{i+\frac{1}{2}} + \Delta t \; u^n_{i+\frac{1}{2}}.$$



Finite volume scheme



The integration of $\partial_t W + \partial_x F(W) = 0$ on the space-time quadrilateral Q gives:

$$\Delta x_i^{n+1,-} W_i^{n+1,-} - \Delta x W_i^n + \Delta t \left(F(W_i^n, W_{i+1}^n) - F(W_{i-1}^n, W_i^n) \right) = 0$$

where $F(W_L, W_R)$ is the Lagrangian flux and Δt satisfies the CFL condition.

Lagrangian flux

We recall that the Lagrangian flux is:

$$\begin{aligned} F(W_i^n, W_{i+1}^n) : &= F(W_{i+\frac{1}{2}}^n) - u_{i+\frac{1}{2}}^n W_{i+\frac{1}{2}}^n, \\ &= (0, p_{i+\frac{1}{2}}^n, u_{i+\frac{1}{2}}^n p_{i+\frac{1}{2}}^n, 0)^T. \end{aligned}$$

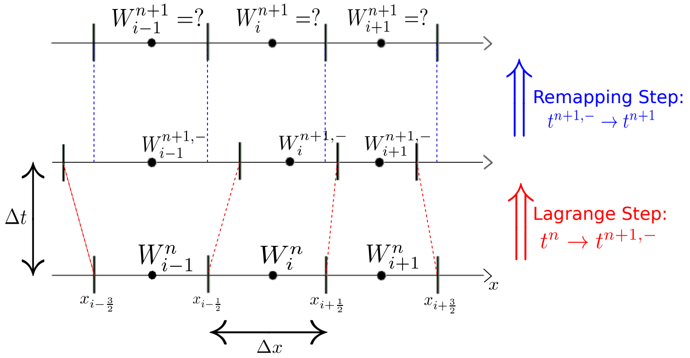
For computing the $i + 1/2$ quantities, we can use an exact Riemann solver or the acoustic Riemann solver [**?**] given by

$$\begin{aligned} u_{i+\frac{1}{2}}^n &= \frac{u_i^n + u_{i+1}^n}{2} - \frac{1}{2\widetilde{\rho c}}(p_{i+1}^n - p_i^n), \\ p_{i+\frac{1}{2}}^n &= \frac{p_i^n + p_{i+1}^n}{2} - \frac{\widetilde{\rho c}}{2}(u_{i+1}^n - u_i^n), \end{aligned}$$

where $\widetilde{\rho c} = \sqrt{\max(\rho_L c_L^2, \rho_R c_R^2)\min(\rho_L, \rho_R)}$.

Remapping step

- The Lagrangian step is done: we have $W_i^{n+1,-}$.
- Problem: how to do the projection to go back to the original grid?



Classical projection: averaging projection

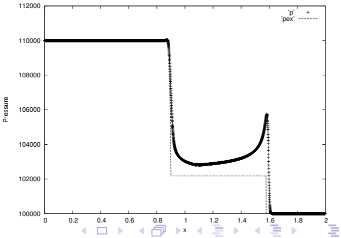
We take:

$$W_i^{n+1} = W_i^{n+1,-} - \frac{\Delta t}{\Delta x} \left(\max(u_{i-\frac{1}{2}},0)(W_i^{n+1,-} - W_{i-1}^{n+1,-}) + \min(u_{i+\frac{1}{2}},0)(W_{i+1}^{n+1,-} - W_i^{n+1,-}) \right).$$

Problem: We consider

Quantities	Left	Right
$\rho(kg.m^{-3})$	10	1
$u(m.s^{-1})$	50	50
$p(Pa)$	1.1e5	1e5
φ	1	0
γ	1.4	1.1
π	0	0

We obtain oscillations on the pressure:



Glimm projection

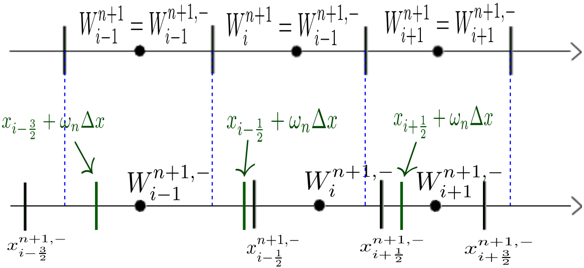
We choose a random point in the cells. According to the position of this point, we pick-up the corresponding value in the Lagrangian mesh.

More precisely, let ω_n be an random number $\in [0;1[$ and we take (see [**?**]):

$$W_i^{n+1} = \begin{cases} W_{i-1}^{n+1,-}, \ \textit{if } x_{i-\frac{1}{2}} + \omega_n \Delta x < x_{i-\frac{1}{2}}^{n+1,-}, \\ W_i^{n+1,-}, \ \textit{if } x_{i-\frac{1}{2}}^{n+1,-} \leq x_{i-\frac{1}{2}} + \omega_n \Delta x \leq x_{i+\frac{1}{2}}^{n+1,-}, \\ W_{i+1}^{n+1,-}, \ \textit{if } x_{i-\frac{1}{2}} + \omega_n \Delta x > x_{i+\frac{1}{2}}^{n+1,-}. \end{cases}$$

Glimm projection

Illustration: We pick up a random number $\omega_n \in [0;1[$.



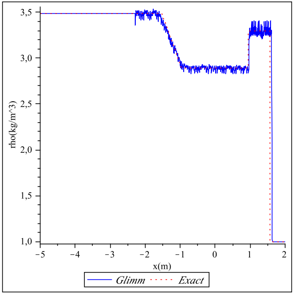
- $x_{i-\frac{3}{2}}^{n+1,-} \leq x_{i-\frac{3}{2}} + \omega_n \Delta x \leq x_{i+\frac{3}{2}}^{n+1,-} \Rightarrow W_{i-1}^{n+1} = W_{i-1}^{n+1,-}$,
- $x_{i-\frac{1}{2}} + \omega_n \Delta x < x_{i-\frac{1}{2}}^{n+1,-} \Rightarrow W_{i-1}^{n+1} = W_{i-1}^{n+1,-}$,
- $x_{i+\frac{1}{2}}^{n+1,-} \leq x_{i+\frac{1}{2}} + \omega_n \Delta x \leq x_{i+\frac{3}{2}}^{n+1,-} \Rightarrow W_{i-1}^{n+1} = W_{i-1}^{n+1,-}$.

Problem: the resulting scheme does not converge

We consider the following Riemann problem: air shock wave impinging a liquid interface

Quantities	Left .	Right
$\rho(kg.m^{-3})$	3.488	1
$u(m.s^{-1})$	1.13	-1
$p(Pa)$	23.33	2
φ	1	0
γ	1.4	2
π	0	7

The resulting scheme does not converge:



Solution: a mixed projection

We apply the Glimm approach only at the two-fluid interface

- If $(\varphi_{i-1}^n - \frac{1}{2})(\varphi_i^n - \frac{1}{2}) < 0$ or $(\varphi_i^n - \frac{1}{2})(\varphi_{i+1}^n - \frac{1}{2}) < 0$,
  We take a random number $\omega_n \in [0, 1[$, and we take:

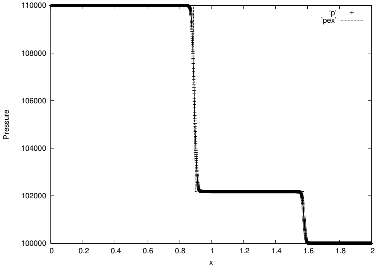
$$W_i^{n+1} = \begin{cases} W_{i-1}^{n+1,-}, \ \textit{if} \ x_{i-\frac{1}{2}} + \omega_n \Delta x < x_{i-\frac{1}{2}}^{n+1,-}, \\ W_i^{n+1,-}, \ \textit{if} \ x_{i-\frac{1}{2}}^{n+1,-} \le x_{i-\frac{1}{2}} + \omega_n \Delta x \le x_{i+\frac{1}{2}}^{n+1,-}, \\ W_{i+1}^{n+1,-}, \ \textit{if} \ x_{i-\frac{1}{2}} + \omega_n \Delta x > x_{i+\frac{1}{2}}^{n+1,-}, \end{cases}$$

- else,

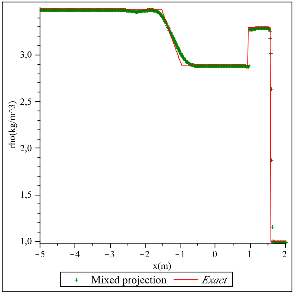
$$W_i^{n+1} = W_i^{n+1,-} - \frac{\Delta t}{\Delta x} \quad (\max(u_{i-\frac{1}{2}}, 0)(W_i^{n+1,-} - W_{i-1}^{n+1,-}) + \min(u_{i+\frac{1}{2}}, 0)(W_{i+1}^{n+1,-} - W_i^{n+1,-})).$$

Results obtained with the mixed projection

There is no oscillations on pressure where the averaging projection failed.

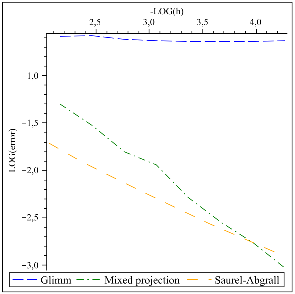


We have no oscillation on the density where the Glimm projection failed:



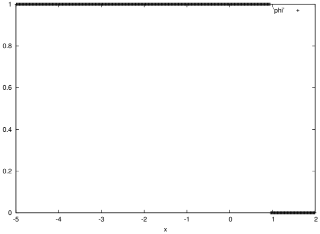
Convergence of the mixed projection

We observe the numerical convergence in the L^1-norm.
The convergence rate is approximately 0.5 for the Saurel-Abgrall approach and 0.8 for the mixed projection.

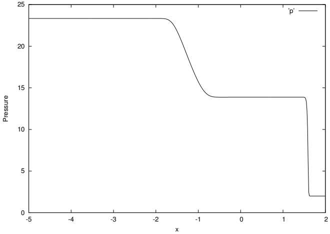


Properties of the scheme

- There is no diffusion on φ: if $\varphi \in \{0, 1\}$, this property is exactly preserved at any time.



- There is no velocity and pressure oscillations at interface:



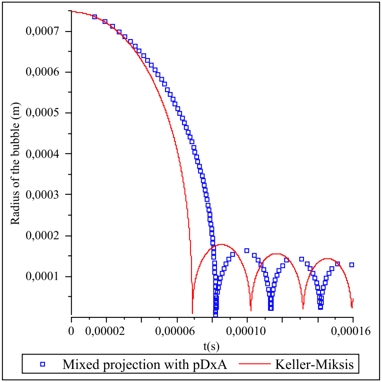
Spherical bubble

In spherical coordinates, the model becomes

$$\begin{aligned}
\partial_t(A\rho) + \partial_x(A\rho u) &= 0, & (1)\\
\partial_t(A\rho u) + \partial_x(A(\rho u^2 + p)) &= pA'(x), \\
\partial_t(A\rho E) + \partial_x(A(\rho E + p)u) &= 0, \\
\partial_t(A\rho\varphi) + \partial_x(A\rho\varphi u) &= 0, & (2)
\end{aligned}$$

where $A(x) = x^2$ appears because of the spherical symmetry. We can compute bubble oscillations with this model and compare it to the Keller-Miksis model.

Preliminary numerical results



Model

We consider the 2D Euler equations:

$$\begin{aligned}
\partial_t(\rho) + \partial_x(\rho u) + \partial_y(\rho v) &= 0, \\
\partial_t(\rho u) + \partial_x(\rho u^2 + p) + \partial_y(\rho u v) &= 0, \\
\partial_t(\rho v) + \partial_x(\rho u v) + \partial_y(\rho v^2 + p) &= 0, \\
\partial_t(\rho E) + \partial_x((\rho E + p)u) + \partial_y((\rho E + p)v) &= 0, \\
\partial_t(\rho \varphi) + \partial_x(\rho \varphi u) + \partial_y(\rho \varphi v) &= 0,
\end{aligned}$$

where ρ is the density, u the x-velocity, v is the y-velocity, E the total energy, φ the fraction of mass of gas and p satisfies the stiffened gas pressure law:

$$p(\rho, e, \varphi) = (\gamma(\varphi) - 1)\rho e - \gamma(\varphi)\pi(\varphi),$$

where

$$e = E - (u^2 + v^2)/2.$$

Model

We can write the system in the conservative form:

$$\partial_t W + \partial_x F(W) + \partial_y G(W) = 0,$$

where the vector of conservative variables is:

$$W = (\rho, \rho u, \rho v, \rho E, \rho\varphi)^T,$$

and the conservative fluxes are:

$$F(W) = (\rho u, \rho u^2 + p, \rho uv, (\rho E + p)u, \rho\varphi u)^T,$$
$$G(W) = (\rho v, \rho uv, \rho v^2 + p, (\rho E + p)v, \rho\varphi v)^T.$$

Splitting

For solving

$$\begin{cases} \partial_t W + \partial_x F(W) + \partial_y G(W) = 0, \\ W(x, y, t = 0) = W_0(x, y), \end{cases}$$

between time $t = 0$ and $t = \Delta t$, we use dimensional splitting.

- Firstly, we solve

$$\begin{cases} \partial_t W + \partial_x F(W) = 0, \\ W(x, y, t = 0) = W_0(x, y), \end{cases}$$

  between time $t = 0$ and $t = \Delta t$, we obtain W_1.

- Secondly, we solve

$$\begin{cases} \partial_t W + \partial_y G(W) = 0, \\ W(x, y, t = 0) = W_1(x, y), \end{cases}$$

  between time $t = 0$ and $t = \Delta t$.

Properties of the scheme

The constructed scheme has the following properties:

- if at initial time the x-velocity u, the y-velocity v and the pressure p are constant, this property is preserved at any time.
- if at initial time the fraction of mass of gas φ takes only the two values 0 or 1, this property is exactly preserved at any time.

What is a GPU?

A modern Graphics Processing Unit (GPU) is made of:

- Global memory (typically 1 Gb)
- Compute units (typically 27).

Each compute unit is made of:

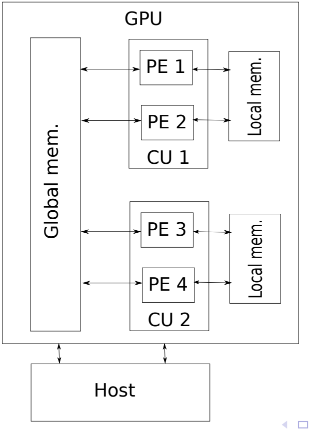
- Processing elements (typically 8).
- Local memory (typically 16 kb)

The same program can be executed on all the processing elements at the same time.

- All the processing elements have access to the global memory
- The processing elements have only access to the local memory of their compute unit.
- The access to the global memory is slow while the access to the local memory is fast.

Example of a GPU

A (virtual) GPU with 2 Compute Units and 4 Processing Elements



OpenCL

- OpenCL means "Open Computing Language". It includes:
  - A library of C functions, called from the host, in order to drive the GPU.
  - A C-like language for writing the kernels that will be executed on the processing elements.
- Virtually, it allows to have as many compute units (work-groups) and processing elements (work-items) as needed.
- The threads are sent to the GPU thanks to a mechanism of command queues on the real compute units and processing elements.
- Portable: the same program can run on a multicore CPU and a GPU. It is also possible to manage several devices in the same program.

Implementation in 1D

- Initialization: we initialize the data on the CPU and we send all the data to the GPU.
- Time step:
  - Computing $W_i^{n+1,-}$: we associate to each cell of the grid one processor (work-item). We compute the fluxes and update the time step for the next time step.
  - We wait that all processors have finished.
  - Update: we apply the mixed projection to obtain W_i^{n+1}.

  $\rightarrow$ We do that while $t < t_{final}$.
- We send all the data to the CPU for post-processing.

Method for the 2D

We want to use our 1D algorithm.
→ We use the rotationnal invariance of the Euler equations.

As

$$F(W) = (\rho u, \rho u^2 + p, \rho uv, (\rho E + p)u, \rho\varphi u)^T,$$
$$G(W) = (\rho v, \rho uv, \rho v^2 + p, (\rho E + p)v, \rho\varphi v)^T,$$

if we note $\tilde{W} = (\rho, \rho v, \rho u, \rho E, \rho\varphi)^T$, we have

$$F(\tilde{W}) = (\rho v, \rho v^2 + p, \rho uv, (\rho E + p)v, \rho\varphi v)^T.$$

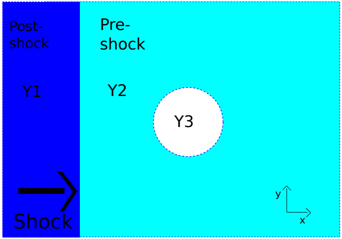
→ We just have to exchange the ρu and ρv place of storage.

Implementation in 2D

- Initialization: we initialize the data on the CPU and we send all the data to the GPU.
- For each time step:
  - We associate to each row of the grid a work-group and to each cell of the row a virtual processor (work-item). We perform the flux computations and projections in the x direction for each work-group.
  - We "transpose": we exchange the ρu and ρv components and we reorganize the data such that the x and y coordinates are exchanged.
  - We perform the flux computations and projections in the y direction for each work group. Thanks to the transposition, the memory access are optimal
  - We transpose to have the correct value in the correct place for the next time step.
- We send all the data to the CPU for post-processing.

Shock-bubble interaction

We consider a shock arriving on a bubble at velocity $\sigma = 415 m.s^{-1}$ (see [?]).



The initial data are:

Quantities	Y1	Y2	Y3
$\rho(kg.m^{-3})$	1.69	1.22	3.86
$u(m.s^{-1})$	113.5	0	0
$v(m.s^{-1})$	0	0	0
$p(Pa)$	1.6e5	1.0e5	1.0e5
φ	0	0	1
γ	1.4	1.4	1.249
π	0	0	0

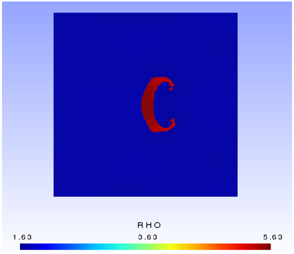
Video

- Number of points of the grid $= 512 \times 512 \simeq 262\,000$.
- Number of unknowns per time step
  $=$ Number of points of the grid $\times 5$.
  $\simeq 1\,310\,000$.

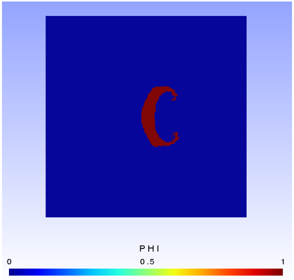
Numerical results

Final time=0.005s.

- The density

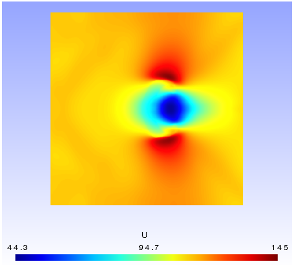


- Masse fraction:

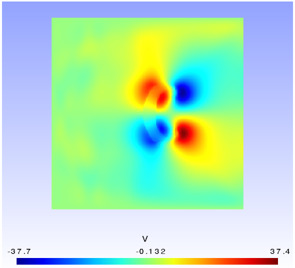


Numerical results

- The x-velocity

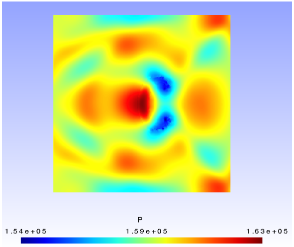


- The y-velocity

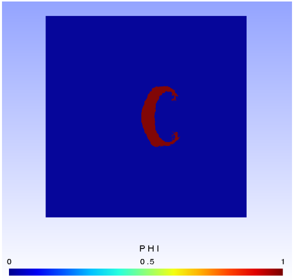


Numerical results

- The pressure

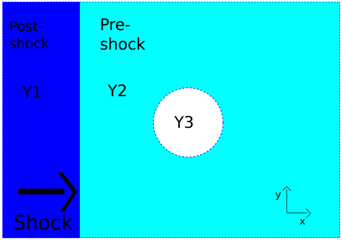


- The interface



Initial condition

We consider a shock that comes to a bubble at velocity $\sigma = 100 m.s^{-1}$.



The initial data are:

Quantities	Y1	Y2	Y3
$\rho(kg.m^{-3})$	1384	1000	1
$u(m.s^{-1})$	27.8	0	0
$v(m.s^{-1})$	0	0	0
$p(Pa)$	$2.87e6$	$1e5$	$1e5$
φ	0	0	1
γ	4.4	4.4	1.4
π	$4.68e5$	$4.68e5$	0

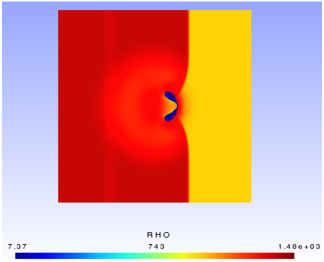
Video

- Number of points of the grid $= 512 \times 512 \simeq 262\,000$.
- Number of unknowns per time step
  $=$ Number of points of the grid $\times 5$.
  $\simeq 1\,310\,000$.

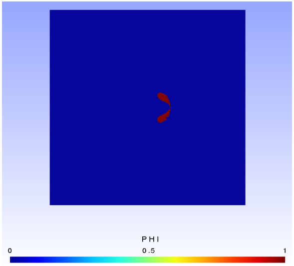
Numerical results

Final time=0.0055s.

- The density

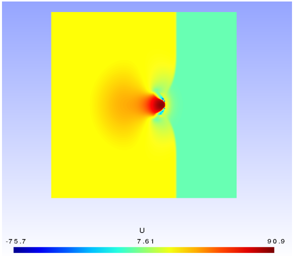


- Mass fraction

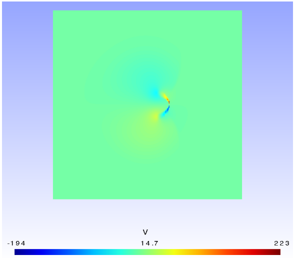


Numerical results

- The x-velocity

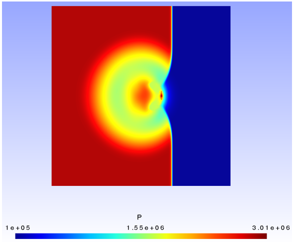


- The y-velocity

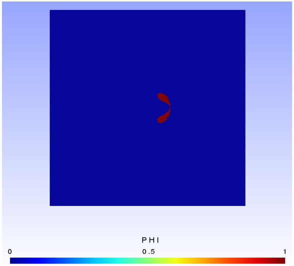


Numerical results

- The pressure



- The interface

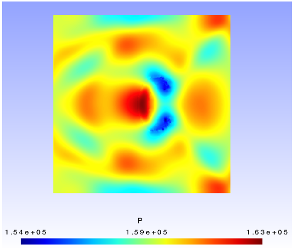


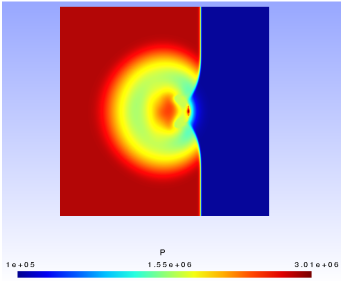
Speedup

The computation was done with a grid of 256×256 points with a the final time is $t_{max} = 0.004$.

	time (s)
AMD Phenom II x4 945 (1 core)	192
AMD Phenom II x4 945 (4 cores)	59
AMD Radeon HD5850	1.43
NVIDIA GTX 460	2.48
NVIDIA Geforce GTX470	0.93

Thank you for your attention!





- B. Després, F. Lagoutière. Numerical resolution of a two-component compressible fluid model with interfaces. Prog. Comp. Fluid Dyn. 7 (6) (2007) 295-310.
- C. Chalons, F. Coquel. Computing material fronts with Lagrange-Projection approach. HYP2010 Proc. http://hal.archives-ouvertes.fr/hal-00548938/fr/.
- S. Kokh, F. Lagoutière. An anti-diffuse numerical scheme for the simulation of interfaces between compressible fluids by means of a five-equation model. Journal of Comp. Phy. 229 (2010) 2773-2809.